Peer-to-Peer Networking and Applications
https://doi.org/10.1007/s12083-021-01206-2

On the robustness of three classes of rateless codes against pollution attacks in P2P networks

Rossano Gaeta[1] · Marco Grangetto[1]

Received: 28 October 2020 / Accepted: 7 June 2021

Abstract

Rateless codes (a.k.a. fountain codes, digital fountain) have found their way in numerous peer-to-peer based applications although their robustness to the so called *pollution attack* has not been deeply investigated because they have been originally devised as a solution for dealing with block erasures and not for block modification. In this paper we provide an analysis of the intrinsic robustness of three rateless codes algorithms, i.e., random linear network codes (RLNC), Luby transform (LT), and band codes (BC) against intentional data modification. By intrinsic robustness we mean the ability of detecting as soon as possible that modification of at least one equation has occurred as well as the possibility a receiver can decode from the set of equations with and without the modified ones. We focus on bare rateless codes where no additional information is added to equations (e.g., tags) or higher level protocol are used (e.g., verification keys to pre-distribute to receivers) to detect and recover from data modification. We consider several scenarios that combine both random and targeted selection of equations to alter and modification of an equation that can either change the rank of the coding matrix or not. Our analysis reveals that a high percentage of attacks *goes undetected* unless a minimum code redundancy is achieved, LT codes are the most fragile in virtually all scenarios, RLNC and BC are quite insensitive to the victim selection and type of alteration of chosen equations and exhibit virtually identical robustness although BC offer a low complexity of the decoding algorithm.

Keywords Rateless codes · Random linear codes · LT codes · Band codes · Data modification attack · Pollution attack

1 Introduction

In the last years, a novel family of asymptotically optimal binary erasure codes, known as rateless codes [9, 11], has gained increasing interest for their flexibility. Rateless codes, as opposed to classical erasure codes, do not require to fix the coding rate a priori, so that a potentially unlimited sequence of equations can be generated. Such an approach was also termed digital fountain or fountain codes [32], the most well known designs being the Luby Transform (LT) [30] and raptor codes [41]. A great deal of research is still going on to devise rateless codes whose decoding complexity and communication overhead are reduced [12, 17, 18, 25, 36, 38, 40]; at the same time, the design of locally repairable codes that allow efficient reconstruction of lost blocks carrying equations is still an active source of interesting proposals [3, 4, 28, 39]. A recent survey [10] nicely summarizes the main developments in rateless codes research.

Oversimplifying, rateless codes partition a data unit into k data fragments of equal size in bits. The data fragments are combined into $n \geq k$ equations each comprising a coded vector (a list of which data fragments are used to define the equation) and a coded fragment (the result of xoring the data fragments indicated by the corresponding coding vector). Each equation is characterized by its degree, i.e., the number of data fragments defined by the coding vector, that is mathematically represented by the degree distribution. The n coding vectors define a $k \times n$ coding matrix $\mathbf{G}$ (also known as generator matrix) and the decoding process can be cast as computing k unknowns (the data fragments composing the data unit) out of a subset of the n equations.

Thanks to their flexibility and simplicity rateless codes have found their way in numerous fields such as distributed storage [2, 31], Wireless Ad Hoc Networks [16], peer-to-peer based applications [43], communications in 5G [37], vehicular networks and Internet of Things applications [13, 34]. In the broad area of peer-to-peer rateless codes have

✉ Rossano Gaeta
rossano.gaeta@unito.it

[1] Dipartimento di Informatica, Università di Torino, Corso Svizzera 189, 10149 Torino, Italy

Published online: 24 June 2021

been exploited in application ranging from multimedia delivery to distributed storage systems [1, 5, 35, 42, 46, 47]. Some of these papers also dealt with security issues. In particular, a line of research has focused on the analysis and containment of the so called pollution attack whereby a set of malicious peers intentionally randomly alter coded fragments to jam the communication and to avoid recovering of the original data unit at the receivers [8, 21, 23]. In these papers malicious peers launch their attack by randomly modifying the coded fragment of victim equations before forwarding them to their neighbor peers.

Unfortunately, the rateless codes' main advantage, i.e. the simplicity of the mechanism used to generate a practically limitless sequence of data fragments robust to erasures, also represents their Achilles' heel: indeed exploiting the same simplicity an attacker can create plausible data fragments capable to pollute the original message. The worst, a few polluted fragments are enough to break completely the decoding process.

Since rateless codes have been originally devised as a solution for dealing with block erasures and not for block modification little or no attention has been devoted to the vulnerability of rateless principle that can be exploited by attackers to modify existing equations or to create new ones on the fly with the objective to jam or pollute the communication and to avoid recovering of the data unit at the receiver.

In this paper we provide an attempt to fill this gap and we investigate the robustness of rateless codes when malicious peers employ more elaborated attack strategies on equations to be forwarded. In particular, we provide a comparison of the intrinsic robustness of three rateless codes algorithms, i.e., random linear network codes (RLNC) [24], Luby Transform (LT) [30], and band codes (BC) [20], against intentional data modification. By intrinsic robustness we mean the ability of receivers to detect as soon as possible that modification of at least one equation has occurred, and that decoding from the set of equations with and without the modified ones is still possible.

We focus on bare rateless codes where no additional information is added to equations, e.g., CRC-like information, or higher level protocol are used, e.g., verification keys pre-distributed to receivers, to detect and recover from data modification. We consider attack strategies where the fragments to modify are selected either randomly or based on some specific feature aiming at increasing their malicious impact. The modification of an equation can either alter the rank of the coding matrix or not.

Our analysis reveals that:

- when the attack load is low, i.e., when only one equation is modified, the attack goes undetected in a high percentage of cases for all rateless code algorithms;
- there exists a minimum number of equations to generate to ensure that detection is always triggered even in a low load attack;
- LT codes are the most fragile with respect to data modification attacks in virtually all scenarios;
- RLNC and BC are almost insensitive to the victim selection and type of alteration of victim equations;
- RLNC and BC exhibit virtually identical robustness although BC offer a low complexity of the decoding algorithm as shown in [20].

The rest of the paper is organized as follows: Section 2 presents the most closely related works dealing with security issues in rateless codes; Section 3 provides the notation used throughout the paper as well as a quick reference to how encoding is carried out by RLNC, LT, and BC; then, Section 4 describes all the details of the scenarios we consider and the definition of robustness indexes we study; Section 5 presents and comments our experimental results; finally, Section 6 summarizes the paper contributions, draws conclusions, and outlines directions for future developments.

For the sake of readability, Table 1 summarizes the key notation used throughout this paper.

2 Related works

As mentioned in the introduction, the rateless coding principle simplifies the procedure to create additional coded fragments that in turn can be exploited by an attacker to modify and inject corrupted data. This kind of problem is also termed pollution or Byzantine attack. Limited attention has been devoted in the scientific literature to such intrinsic vulnerability of the rateless principle: in this section we will briefly review the most closely related research.

Data modification attack has been considered in several paper in the context of peer-to-peer based applications, e.g., [8, 21, 23]. In these works a set of malicious peers intentionally alter coded fragments to jam the communication and to avoid recovering of the original data unit at the receivers by randomly replacing the coded

Table 1 Key notation

	Meaning	Values
rc	rateless code	$\{RLNC, LT, BC\}$
k	# data unit fragments	{32, 256}
n	# equations	$[k+4, 1.5k]$
p	# modified equations	[1, 4]
ch	victim choice strategy	{RANDOM, LOWEST, HIGHEST }
md	modification strategy	{UNM, TOL, TOH}

fragment of victim equations before forwarding them to their neighbor peers. These works only consider this kind of attack and develop identification algorithms to spot malicious peers and secure data communication.

Pollution attack has been also studied in the context of network coding [44], where intermediate nodes may compromise on security. A well known approach to tackle a Byzantine adversary is to add cryptographic functions, e.g. hashes or signatures, to each coded fragments [22, 26, 27]. Besides the additional computational cost and need of an ancillary secure channel to exchange cryptographic keys, these solutions are further complicated by the rateless principle since coded fragments cannot be known and signed in advance as in the case of fixed rate erasure codes. The works in [29, 48] improve these kind of approaches by introducing homomorphic signature that enables intermediate nodes to verify messages without the need of a secure channel for key pre-distribution. Homomorphic authentication is further enhanced in [29] to secure regenerating codes in the context of cloud storage with the additional property of being privacy preserving. In [14] improved key distribution schemes for homomorphic subspace signature for network coding is proposed.

Few works have analyzed the effect of data modification on rateless codes in general, outside the context of a particular application such as network coding or cloud storage. Closer to our work the approach in [33] considers a wireless scenario where one wishes that the legitimate user receives enough fountain packets to decode first as opposed to attackers that want to get the private message. In this case the attacker is a receiver only and cannot inject corrupted data. In [15] the authors propose to counteract the vulnerability of rateless codes to coded fragment modification by designing a specific coding strategy that is resilient to Byzantine attacks when the fraction of corrupted packets is guaranteed to be less than 1/3 of the total coded fragments. As opposed to our study this amounts at using ad-hoc coding/decoding solutions.

3 Background on rateless codes

A rateless encoder partitions a data unit into k data fragments $\mathbf{m} = (m_1, \ldots, m_k)$ of equal size of z bits and then generates n $(n \geq k)$ coded fragments $\mathbf{y} = (y_1, \ldots, y_n)$ according to some coding algorithm rc . Each coded fragment y_i is computed as a linear combination (binary XOR operation) of the data fragments: every coded fragments is associated to a coding equation or coding vector. This latter is a column $k \times 1$ binary vector $\mathbf{g} = (g_1, \ldots, g_k)'$ with $g_j = 1$ if the j-th data fragments m_j is selected for combination, and $g_j = 0$ otherwise. Each coding equation is characterized by its *degree* d, i.e., the number of data fragments defined by the coding vector (number of 1s signaled in the vector). The coding vectors of all n coded fragments can be arranged column-wise to form a $k \times n$ coding matrix $\mathbf{G}$ (also known as generator matrix) and any linear block code can be represented as linear mapping $\mathbf{y} = \mathbf{mG}$, through the $k \times n$ coding matrix, with addition and multiplication defined in the binary Galois field. In turn, the decoding process can be cast as computing k unknowns m_i out of a subset of the n coded fragments. For instance, this goal can be achieved by using Gaussian elimination [6] to solve the linear problem $\mathbf{y}_{\mathcal{S}} = \mathbf{mG}_{\mathcal{S}}$, where $\mathbf{y}_{\mathcal{S}}$ represents a subset of the coded fragments indexed by set $\mathcal{S} \subseteq \{1, \ldots, n\}$, and $\mathbf{G}_{\mathcal{S}}$ are the corresponding equations, i.e. columns in the generator matrix. The linear problem can be solved if $\mathbf{G}_{\mathcal{S}}$ comprises exactly k linear independent equations. If n coded fragments are needed to decoding we define the overhead of the code as $\epsilon = n/k - 1$.

In this paper the values of rc we analyze are RLNC [24], LT [30], and BC [20]; in the rest of this section we briefly sketch the algorithms to compute coded fragments $\mathbf{y}$ in each case and we refer the reader to the original papers for full details.

3.1 LT algorithm to compute equations

LT [30] are known as the first codes designed according to the rateless principle while retaining linear decoding complexity and asymptotically optimal property, i.e. $\epsilon \to 0$ for $k \to \infty$. The key to LT success is related to the Robust Soliton distribution $\mu(c, \delta)$ (RSD) from which the degree d of each y_i is sampled (c and δ are two parameters that shape the distribution). The RSD enforces mostly low degree equations (the peak of the RSD is for $d = 2$) and few high degree equations: in particular, it is designed so that it is possible to decode by looking for equations with $d = 1$ (of kind $y_i = m_j$), xoring the data fragments m_j from all remaining equations (that will get degree lowered by 1) and iterating the process.

3.2 RLNC algorithm to compute equations

Before the advent of rateless codes a lot of work has been devoted to the so called network coding, and in particular RLNC [24], where some limitations of classical store and forward approach followed in packed switched networks could be solved by applying linear coding to packets. In this case, a new coded fragment y_i is generated by any random combination of data fragments: it turns out that on average one gets $d = k/2$. As opposed to LT the decoding complexity is high and must be based on Gaussian elimination.

3.3 BC algorithm to compute equations

BC aim at striking a balance between LT and RLNC in terms of complexity [20]. To this end, the random approach is retained only within the so called encoding windows or band, whose size $W < k$ is constrained by design. To encode y_i first the leading edge f of the window is selected, then only data fragments $m_f, \ldots, m_{f+W-1}$ can are combined randomly. To get a functional design one needs to adopt some modifications to the presented basic principle taking into consideration the cases when the encoding band approaches the trailing edge of the coding block, i.e. when $(f + W - 1) > k$. We refer the interested reader to [20] for details. The key point of BC is that the average value of d turns to be $W/2$ and allowing one to control the decoding complexity and overhead. As for RLNC Gaussian elimination decoder is required.

4 Scenario description

We consider a simple abstract scenario composed of:

- a transmitter that encodes a data unit partitioned in k data fragments $\mathbf{m} = (m_1, \ldots, m_k)$ and sends n coded fragments $\mathbf{y} = (y_1, \ldots, y_n)$ according to some coding algorithm rc for rateless codes as described in Section 3. The generation of equations is iterated until $\mathbf{G}$ is full rank, i.e., the n equations include a set of k linearly independent equations from which it is possible to recover the data unit $\mathbf{m}$.
- an attacker that can intentionally modify a subset of p equations ($1 \leq p \leq n$);
- a receiver that is fed with the set of n equations including the modified ones.

The subset of p modified equations can be selected according to three possible strategies ch:

- a random subset of cardinality p is selected as the victim equations of the data modification attack (RANDOM strategy);
- p equations corresponding to those with the highest degree are selected (HIGHEST strategy);
- p equations corresponding to those with the lowest degree are selected (LOWEST strategy).

Please note that this abstract mechanism to select victim equations can be mapped onto different more realistic attack models. For instance, the RANDOM strategy could represent an attacker that has only a partial view of the set of n equations and it is only able to intercept p of them because the remaining $n - p$ follow different paths from data source to destinations. HIGHEST and LOWEST strategies could be employed by a set of malicious storage nodes in a distributed storage system; these attackers collectively share a partial yet larger view of the set of n equations and can select victim equations based on their degree.

Finally, for each choice strategy ch the modification of an equation can occur in three different ways. The equation degree can change by:

- leaving the coding vector unaltered and by randomly substituting the coded fragment value (UNM);
- replacing the original coding vector by a fake one involving only a single randomly selected data fragment (TOL). This amounts to setting the degree of the modified equation to 1;
- replacing the original coding vector by a fake one involving all data fragments composing the data unit (TOH). This amounts to setting the degree of the modified equation to k.

The receiver uses the decoding algorithm in [6] for recovering the data unit from the set of n equations. Detection of data unit modification is carried out by a trivial extension of a Gauss elimination algorithm where at each step a check is performed to verify if an inconsistency in the progressive solution of a redundant system of linear equation is found. Although such algorithm cannot raise an alert when none of the equations has been altered the detection of modification might fail, instead. In the simplest case, a detection failure might occur when all n equations are absolutely necessary for decoding, i.e., when all subsets of $n - 1$ equations do not include a set of k linearly independent equations.

4.1 Robustness indexes

We evaluate robustness of rateless codes along three dimensions by the following indexes:

- detection robustness r_{det} is the probability that detection of modified equations is triggered;
- attack robustness r_{att} is the probability that recover of the original data unit is still possible after removal of modified equations occurs;
- detection earliness e_{det} is the average number of equations processed by the detector before issuing an alert normalized with respect to k. Therefore, it can range between $\frac{1}{k}$ and $\frac{n}{k}$ where smaller values represent earlier detection of a modification attack on a data unit.

The ideally robust rateless code should have the first two indexes as close to 1 as possible and a detection earliness as small as possible.

5 Results

In this section we present the architecture of the test-bed we developed to analyze and characterize the robustness of rateless codes as well as results for robustness indexes as defined in Section 4.1.

5.1 Setup

To evaluate the robustness of rateless codes against data modification we developed C++ prototypes representing our test−bed for the simulation of the interaction among transmitter, attacker, and receiver. In particular, the test−bed includes the following components:

- the generator whose task is to encode $n \geq k$ equations according to the chosen rateless coding algorithm rc;
- the attacker module, that modifies p equations according to choice strategies ch and equation degree strategy md as described in Section 4.
- the decoder that is implemented according to the Gaussian Elimination method proposed in [6]. The decoder consumes one equation at a time to progressively simplify the system of linear equations; it is capable to return an alert to signal that the data unit has been corrupted as discussed in Section 4 as well as the number of equations processed when detection occurs for the first time.

To characterize the robustness of a rateless code algorithm rc we ran 100,000 independent trials for each combination of system parameters represented by the 6−tuple (rc, n, k, p, ch, md) for the values summarized in Table 1. Each trial is organized as follows: n equations are generated that define a full rank coding matrix $\mathbf{G}$ and p of them are chosen according to ch and modified according to md. The decoder is invoked on the set of n equations to verify if decoding is possible and a data modification alert is possibly triggered by the detector. Finally, the decoder is newly invoked on the set of $n - p$ unmodified equations to verify if recover of the original data unit is still possible.

Robustness indexes defined in Section 4.1 are estimated as the fraction of trials that:

- trigger the detector (r_{det});
- allow one to recover the original data unit from the $n - p$ unmodified equations (r_{att}).

The index e_{det} is obtained as the average value of the number of equations processed by the detector before issuing an alert normalized with respect to k. Finally, for LT we used a Robust Soliton distribution $\mu(c, \delta)$ where $c = 0.05$ and $\delta = 0.01$ and for BC we considered a band size $W = \frac{3k}{4}$.

5.2 Impact on detection robustness r_{det}

Detection robustness r_{det} represents the probability the detector triggers an alert to signal that data modification has occurred and that the data unit recovered by the decoder is not valid. It is an increasing function of both normalized redundancy $\frac{n}{k}$ and the number of modified equations p. For $\frac{n}{k} = 1.5$ detection robustness approaches 1 for all scenarios, i.e., detection is always triggered; the same result is obtained when $p = 4$ in all other settings. Figure 1 (left graph) depicts r_{det} as a function of normalized redundancy $\frac{n}{k}$ for scenarios $(rc = *, k = 32, n = *, p = 1, ch = \text{RANDOM}, md = \text{TOL})$ where a single equation is modified. Analogously, Figure 1 (right graph) displays results as a function of the number of modified equations p for scenarios $(rc = *, k = 32, n = k + 4, p = *, ch = \text{RANDOM}, md = \text{TOL})$. It can be noted that a stealth attack in a low redundancy setting is by far the most insidious scenario since the probability the attack goes undetected (which is equal to $1 - r_{det}$) is unacceptably high. Indeed, about $3 - 4\%$ of the attacks where only 1 equation is modified *goes undetected for all rateless code algorithms*. Furthermore, Table 2 shows that RLNC and BC are quite insensitive to how victim equations are chosen and modified while LT codes suffer the worst performance that *do depend* on ch and md. In particular, up to about $12 - 13\%$ of data modification attacks goes undetected when the ch =LOWEST equations are selected and only the value of the coded fragment is randomly altered, i.e., when md =UNM. The attack strategy where the coding vector is unaltered and only the value of the coded fragment is changed has also a harder impact on RLNC and BC: in this case about 7% of attacks is missed by the detector.

Results in Table 2 also show that conclusions are independent of the value of k. Indeed, we obtained very similar results for all values of k tested in the range [32, 256]; therefore, to avoid cluttering graphs and tables in the sequel we only present results for the case $k = 32$.

5.3 Impact on attack robustness r_{att}

Attack robustness r_{att} represents the probability that recovering the original clean data unit is still possible after removal of the modified equations. It represents a lower bound on the actual recovery capability since we are implicitly assuming that a perfect identifier of modified equations is available.

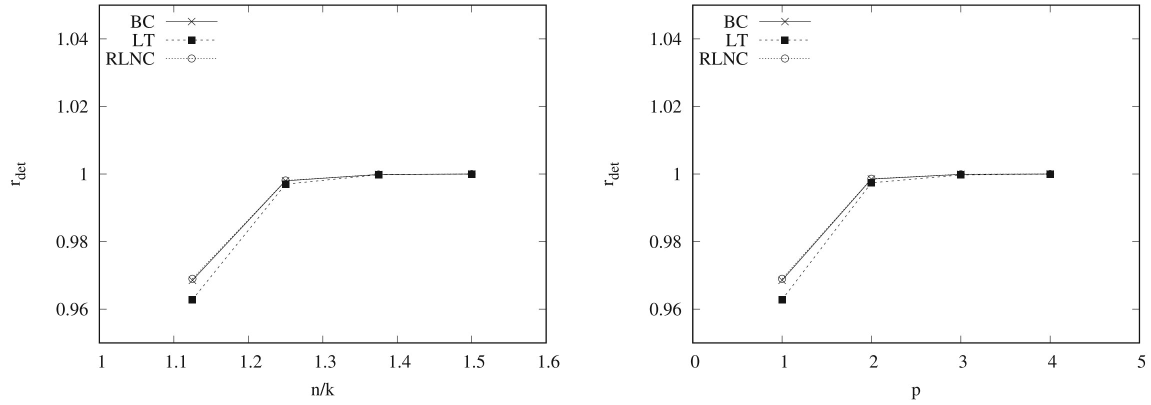

Fig. 1 Detection robustness r_{det} as a function of normalized redundancy $\frac{n}{k}$ for scenarios $(rc = *, k = 32, n = *, p = 1, ch =$ RANDOM, $md =$ TOL$)$ (left) and as a function of the number of modified equations p for scenarios $(rc = *, k = 32, n = k + 4, p = *, ch =$ RANDOM, $md =$ TOL$)$ (right)

Figure 2 (left graph) depicts r_{att} as a function of normalized redundancy $\frac{n}{k}$ for scenarios $(rc = *, k = 32, n = *, p = 1, ch =$ RANDOM, $md =$ TOL$)$ (where a single equation is modified). Figure 2 (right graph) displays results as a function of the number of modified equations p for scenarios $(rc = *, k = 32, n = k + 4, p = *, ch =$ RANDOM, $md =$ TOL$)$.

We observe r_{att} is an increasing function of normalized redundancy $\frac{n}{k}$ while it is markedly decreasing as a function of the number of modified equations p. For $\frac{n}{k} = 1.5$ attack robustness approaches 1 for all scenarios, i.e., recovery is always possible.

We also note that in a low redundancy scenario the possibility of recovery quickly deteriorates as the data modification attack gets harsher; for $p = 4$ it is almost impossible for all rateless code algorithms to avoid disruption of the data unit.

Again, Table 3 shows that while RLNC and BC are quite insensitive to how victim equations are chosen and modified, LT codes suffer the worst performance among all rateless code algorithms. In particular, when only one equation is modified LT is not able to survive a data modification attack in about 12% of the cases when lowest degree equations are selected as victims regardless the actual modification choice.

5.4 Impact on detection earliness e_{det}

Detection earliness e_{det} is the average number of equations processed by the detector before issuing an alert normalized with respect to k. It ranges in the interval $[\frac{1}{k}, \frac{n}{k}]$ where smaller values represent more prompt detection of a modification attack on a data unit. Figure 3 (left graph) depicts e_{det} as a function of normalized redundancy $\frac{n}{k}$ for scenarios $(rc = *, k = 32, n = *, p = 1, ch =$ RANDOM, $md =$ TOL$)$. It can be noted that the average number of equations to be processed by the detector before issuing an alert is almost always greater than or equal to k. In this case LT allow for the earliest detection among all three codes.

Figure 3 (right graph) shows e_{det} as a function of the number of modified equations p for scenarios $(rc = *, k = 32, n = k + 4, p = *, ch =$ RANDOM, $md =$ TOL$)$. We observe that the higher the number of modified equations

Table 2 Detection robustness r_{det} as a function of choice strategies ch and degree modification md for $n = k + 4$, and $p = 1$

k	rc	RANDOM			LOWEST			HIGHEST		
		UNM	TOL	TOH	UNM	TOL	TOH	UNM	TOL	TOH
32	RLNC	.937	.968	.968	.937	.968	.968	.937	.968	.968
	LT	.901	.962	.952	.883	.960	.935	.922	.964	.962
	BC	.935	.968	.968	.935	.969	.967	.936	.968	.968
256	RLNC	.939	.968	.968	.935	.969	.966	.941	.967	.973
	LT	.870	.949	.935	.870	.942	.930	.897	.954	.950
	BC	.936	.965	.966	.937	.967	.970	.936	.972	.967

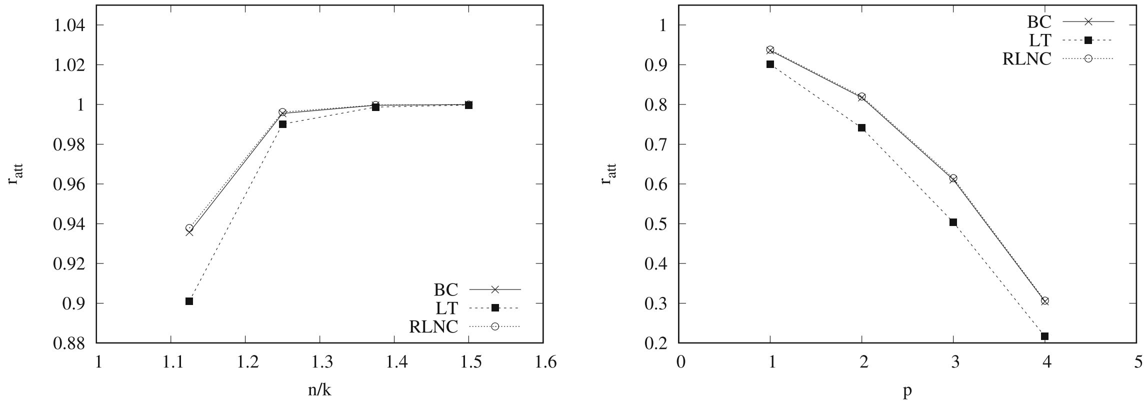

Fig. 2 Attack robustness r_{att} as a function of normalized redundancy $\frac{n}{k}$ for scenarios ($rc = *, k = 32, n = *, p = 1, ch =$ RANDOM, $md =$ TOL) (left) and as a function of the number of modified equations p for scenarios ($rc = *; k = 32; n = k + 4; p = *; ch =$ RANDOM, $md =$ TOL) (right)

the earlier detection occurs. Again, LT reveals as the most reactive rateless code in this comparison.

Finally, Table 4 shows that reactivity of detection is quite similar for all rateless codes in all scenarios. When the number of modified equations is higher we observe that regardless the rateless code algorithm and the choice strategy of victim equations the data modification attack that triggers the most reactive detection is $md =$TOH. In all cases results confirm that LT is by far the most reactive rateless code; they also show that RLNC and BC exhibit almost identical results.

5.5 Discussion

A comprehensive and rigorous theoretical analysis of the robustness of codes we considered against all kind of attacks could be rather complex. Nevertheless, results characterizing detection robustness r_{det} and attack robustness r_{att} for scenarios ($rc = *, k = *, n = k + 4, p = 1, ch =$ RANDOM, $md =$ UNM) (Tables 2 and 3) could be explained by resorting to the analysis of the probability that coding matrix $\mathbf{G}$ is full rank k when n equations are available to the decoder (we denote it as $p_{fr}(n, k)$). Analytical expressions of $p_{fr}(n, k)$ for RLNC are given in [45, 49] while numerical solutions for LT is provided by [19]. Unfortunately, there is no analytical or numerical solution for BC although simulations show that for band size $W > \sqrt{k}$ results are close to those for RLNC.

In the experiments, we generated equations according to the chosen coding algorithm and we selected a set whose size is n such that it defines a full rank k coding matrix $\mathbf{G}$. Under random selection of a victim equation ($ch =$ RANDOM) and modification of its coded fragment ($md =$ UNM) the rank of coding matrix $\mathbf{G}$ remains unaltered. In this case, modification of one equation just results in the redefinition of a new system of linear equations to be solved. If none of the n subsets of equations whose cardinality is equal to $n - 1$ defines a rank k coding matrix then it is inevitable not to detect any change in the entire set of n equations encoding the original data unit. This event occurs with probability $1 - \frac{p_{fr}(n-1)}{p_{fr}(n)}$, i.e., the probability that decoding is possible only with no less than n equations. We implemented algorithms in [19, 45, 49] to compute $\frac{p_{fr}(n-1)}{p_{fr}(n)}$ for $k = 32$: we obtained 0.9375 and 0.9141 for RLNC and

Table 3 Attack robustness r_{att} as a function of choice strategies ch and degree modification md for $k = 32$ and $n = k + 4$

p	rc	RANDOM			LOWEST			HIGHEST		
		UNM	TOL	TOH	UNM	TOL	TOH	UNM	TOL	TOH
1	RLNC	.937	.937	.937	.937	.937	.937	.937	.937	.937
	LT	.901	.901	.901	.883	.882	.883	.922	.922	.922
	BC	.935	.935	.935	.935	.935	.935	.936	.936	.936
4	RLNC	.306	.306	.306	.307	.308	.307	.308	.308	.308
	LT	.216	.216	.216	.166	.166	.166	.267	.267	.267
	BC	.304	.304	.304	.303	.303	.303	.305	.304	.305

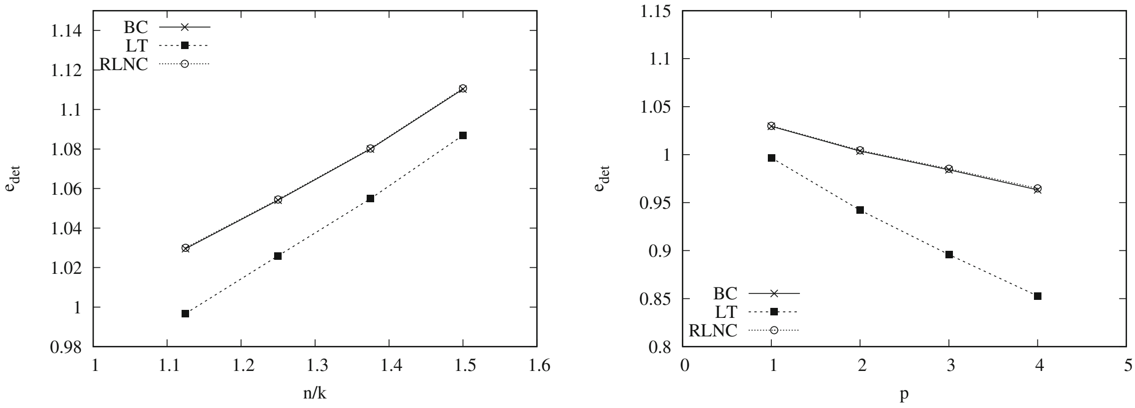

Fig. 3 Detection earliness e_{det} as a function of normalized redundancy $\frac{n}{k}$ for scenarios ($rc = *, k = 32, n = *, p = 1, ch =$ RANDOM, $md =$ TOL) (left) and as a function of the number of modified equations p for scenarios ($rc = *, k = 32, n = k + 4, p = *, ch =$ RANDOM, $md =$ TOL) (right)

LT, respectively. These values are in excellent agreement with those we show in Table 2. We also computed the same quantities for $k = 256$: we obtained 0.9375 for RLNC (in excellent agreement with results in Table 2) but we could not do the same for LT since the numerical approach in [19] becomes unstable for $k > 64$.

A similar analysis can also explain results for attack robustness r_{att} in the same scenarios. Indeed, $\frac{p_{fr}(n-1)}{p_{fr}(n)}$ we previously computed for detection robustness r_{det} also represents the probability that decoding is still possible even if we remove the randomly selected victim equation. Also in this case, it can be noted that they are in excellent agreement with those presented in Table 3. More generally, when p victim equations are removed from the set of n equations, attack robustness r_{att} results can be predicted by considering probability $\frac{p_{fr}(n-p)}{p_{fr}(n)}$. For instance, for $k = 32$ when $p = 4$ we compute 0.3076 and 0.2146 for RLNC and LT, respectively. These values are almost identical to those presented in Table 3.

Unfortunately, both degree targeted selection of victim equations and its degree modification can modify the rank of coding matrix $\mathbf{G}$ making theoretical analysis of results a lot more complex.

6 Conclusions and future works

Rateless codes have found their way in numerous peer-to-peer base applications but since they have been originally devised as a solution for dealing with block erasures and not for block modification little or no attention has been devoted to the analysis of their robustness when malicious peers modify existing equations on the fly with the goal to avoid recovering of the original data unit at the receiver.

In this paper we provided an attempt to fill this gap and we investigated the robustness of rateless codes when malicious peers employ more elaborated attack strategies on equations to be forwarded. We analyzed by means of a full C++ implementation of a test−bed composed of encoders, decoders, and detector the intrinsic robustness of three rateless code algorithms against intentional modification of equations. We focused on bare rateless codes where no additional information is added to equations or higher level protocol are used, e.g., verification keys to pre-distribute to receivers to detect and recover from data modification. We formalized the term *intrinsic robustness* by three indexes: detection robustness, attack robustness, and detection earliness that we evaluated in a large number

Table 4 Detection earliness e_{det} as a function of choice strategies ch and degree modification md for $k = 32$ and $n = k + 4$

p	rc	RANDOM			LOWEST			HIGHEST		
		UNM	TOL	TOH	UNM	TOL	TOH	UNM	TOL	TOH
1	RLNC	1.030	1.030	1.029	1.028	1.027	1.027	1.028	1.027	1.027
	LT	1.028	0.996	1.023	1.035	0.981	1.034	0.999	0.997	1.012
	BC	1.030	1.029	1.030	1.027	1.026	1.027	1.027	1.026	1.027
4	RLNC	.989	.964	.461	.988	.959	.398	.988	.959	.398
	LT	.974	.852	.458	1.00	.751	.238	.940	.897	.241
	BC	.989	.963	.461	.988	.956	.388	.988	.957	.388

of scenarios for all combinations of code parameters and attack strategies.

Our analysis provided several interesting observations: a high percentage of attacks *goes undetected* unless a minimum code redundancy is achieved, LT codes are the most fragile with respect to data modification attacks in virtually all scenarios, RLNC and BC are quite insensitive to the victim selection and type of alteration of victim equations and exhibit virtually identical robustness although BC offer a low complexity of the decoding algorithm [20].

Future developments include the analysis of a detector based on optimal intermediate decoders for rateless codes [7], the extension of the test−bed with a non−ideal algorithm to identify modified equations [2] and the analysis of additional rateless codes and type of attacks. e.g., those in [15].

Funding Open access funding provided by Università degli Studi di Torino within the CRUI-CARE Agreement.

References

1. Ahmad S, Bouras C, Buyukkaya E, Dawood M, Hamzaoui R, Kapoulas V, Papazois A, Simon G (2018) Peer-to-peer live video streaming with rateless codes for massively multiplayer online games. Peer-to-Peer Networking and Applications 11(1):44–62
2. Anglano C, Gaeta R, Grangetto M (2017) Securing coding-based cloud storage against pollution attacks. IEEE Trans Parallel Distrib Sys 28(5):1457–1469
3. Asteris M, Dimakis AG (2014) Repairable fountain codes. IEEE J Sel Areas Commun 32(5):1037–1047
4. Baik J, Suh Y, Shin M, Kim S, Kim J (2020) Locality-improved repairable fountain codes for distributed storage systems. In: ICC 2020 - 2020 IEEE International Conference on Communications (ICC), pp 1–6
5. Bioglio V, Gaeta R, Grangetto M, Sereno M (2013) Rateless codes and random walksfor p2p resource discovery in grids. IEEE Trans Parallel Distrib Sys 25(4):1014–1023
6. Bioglio V, Grangetto M, Gaeta R, Sereno M (2009) On the fly gaussian elimination for lt codes. IEEE Commun Lett 13(12):953–955
7. Bioglio V, Grangetto M, Gaeta R, Sereno M (2011) An optimal partial decoding algorithm for rateless codes. In: 2011 IEEE International symposium on information theory proceedings, pp. 2731–2735
8. Buttyán L., Czap L, Vajda I (2010) Detection and recovery from pollution attacks in coding-based distributed storage schemes. IEEE Transactions on Dependable and Secure Computing 8(6):824–838
9. Byers JW, Luby M, Mitzenmacher M (2002) A digital fountain approach to asynchronous reliable multicast. IEEE J Sel Areas Commun 20(8):1528–1540
10. Byers JW, Luby M, Mitzenmacher M (2019) A digital fountain retrospective. SIGCOMM Comput Commun Rev 49(5):82–85
11. Byers JW, Luby M, Mitzenmacher M, Rege A (1998) A digital fountain approach to reliable distribution of bulk data. In: Proceedings of the ACM SIGCOMM '98 conference on applications, technologies, architectures, and protocols for computer communication, SIGCOMM '98, pp. 56–67. New York, NY, USA
12. Cassuto Y, Shokrollahi A (2015) Online fountain codes with low overhead. IEEE Trans Inf Theory 61(6):3137–3149
13. Chen X, Jia R (2018) Exploiting rateless coding for massive access. IEEE Trans Veh Technol 67(11):11253–11257
14. Cheng C, Lee J, Jiang T, Takagi T (2016) Security analysis and improvements on two homomorphic authentication schemes for network coding. IEEE Transactions on Information Forensics and Security 11(5):993–1002
15. Cohen A, Dolev S, Tzachar N (2013) Efficient and universal corruption resilient fountain codes. IEEE Trans Commun 61(10):4058–4066
16. Deng N, Haenggi M (2019) Delay characterization of rateless codes in wireless ad hoc networks. In: ICC 2019 - 2019 IEEE International Conference on Communications (ICC), pp. 1–6
17. Deng N, Haenggi M (2019) The end-to-end performance of rateless codes in poisson bipolar and cellular networks. IEEE Trans Commun 67(11):8072–8085
18. Feng S, Yang J (2019) Age-optimal transmission of rateless codes in an erasure channel. In: ICC 2019 - 2019 IEEE International Conference on Communications (ICC), pp. 1–6
19. Lu F, Foh CH, Cai J, Chia L (2009) Lt codes decoding: Design and analysis. In: IEEE International symposium on information theory, pp. 2492–2496
20. Fiandrotti A, Bioglio V, Grangetto M, Gaeta R, Magli E (2014) Band codes for energy-efficient network coding with application to p2p mobile streaming. IEEE Trans Multimed 16(2):521–532
21. Fiandrotti A, Gaeta R, Grangetto M (2019) Securing network coding architectures against pollution attacks with band codes. IEEE Trans Inf Forensics Security 14(3):730–742
22. Gkantsidis C, Rodriguez Rodriguez P (2006) Cooperative security for network coding file distribution. In: Proceedings IEEE INFOCOM 2006. 25TH IEEE international conference on computer communications, pp. 1–13
23. He H, Li R, Xu Z, Xiao W (2014) An efficient ecc-based mechanism for securing network coding-based p2p content distribution. Peer-to-Peer Networking and Applications 7(4):572–589
24. Ho T, Medard M, Koetter R, Karger DR, Effros M, Shi J, Leong B (2006) A random linear network coding approach to multicast. IEEE Trans Inf Theory 52(10):4413–4430
25. Huang J, Fei Z, Cao C, Xiao M (2020) Design and analysis of online fountain codes for intermediate performance. IEEE Trans Commun: 1–1
26. Kehdi E, Li B (2009) Null keys: Limiting malicious attacks via null space properties of network coding. In: IEEE INFOCOM 2009, pp 1224–1232
27. Krohn MN, Freedman MJ, Mazieres D (2004) On-the-fly verification of rateless erasure codes for efficient content distribution. In: IEEE Symposium on Security and Privacy, 2004. Proceedings. 2004, pp 226–240

28. Kumar S, Rosnes E, Graell i Amat A (2016) Secure repairable fountain codes. IEEE Commun Lett 20(8):1491–1494
29. Liu J, Huang K, Rong H, Wang H, Xian M (2015) Privacy-preserving public auditing for regenerating-code-based cloud storage. IEEE Trans Inf Forensics Security 10(7):1513–1528
30. Luby M (2002) LT codes. In: IEEE FOCS, pp. 271–280
31. Luby M, Padovani R, Richardson TJ, Minder L, Aggarwal P (2019) Liquid cloud storage. ACM Transactions on Storage 15(1)
32. MacKay D (2005) Fountain codes. IEE Proceedings - Communications 152 (6):1062–1068
33. Niu H, Iwai M, Sezaki K, Sun L, Du Q (2014) Exploiting fountain codes for secure wireless delivery. IEEE Commun Lett 18(5):777–780
34. Pan Z, Lei J, Liu W, Luo J, Tang C (2019) Grant-free rateless scma for cellular internet of things networks. IEEE Access 7:147954–147961
35. Park GS, Song H (2016) A novel hybrid p2p and cloud storage system for retrievability and privacy enhancement. Peer-to-Peer Networking and Applications 9(2):299–312
36. Rafie Borujeny, Ardakani RM (2016) A new class of rateless codes based on reed–solomon codes. IEEE Trans Commun 64(1):49–58
37. Rajanna A, Dettmann CP (2020) Rate statistics in cellular downlink: A per-user analysis of rateless coded transmission. IEEE Commun Lett 24(6):1221–1225
38. Schnelling C, Rothe M, Mathar R, Schmeink A (2018) Rateless codes based on punctured polar codes. In: 2018 15th International Symposium on Wireless Communication Systems (ISWCS), pp. 1–5
39. Selvakumar R et al (2020) Reliable and secure data communication in wireless sensor networks using optimal locally recoverable codes. Peer-to-Peer Networking and Applications 13(3)
40. Shi P, Wang Z, Li D, Xiang W (2020) Zigzag decodable online fountain codes with high intermediate symbol recovery rates. IEEE Trans Commun: 1–1
41. Shokrollahi A (2006) Raptor codes. IEEE Transactions on Information Theory 52(6):2551–2567
42. Su Z, Wang F, Daigle J, Wang H, Shan T (2015) Raptorqp2p: Maximize the performance of p2p file distribution with raptorq coding. In: 2015 IEEE International Conference on Communications (ICC)
43. Thomos N, Frossard P (2010) Network coding of rateless video in streaming overlays. IEEE Trans Circuits Sys Video Technol 20(12):1834–1847
44. Tong W, Zhong S (2016) A unified resource allocation framework for defending against pollution attacks in wireless network coding systems. IEEE Transactions on Information Forensics and Security 11(10):2255–2267
45. Trullols-Cruces O, Barcelo-Ordinas JM, Fiore M (2010) Exact decoding probability under random linear network coding. IEEE Commun Lett 15(1):67–69
46. Westphal C (2014) A stable fountain code mechanism for peer-to-peer content distribution. In: IEEE INFOCOM
47. Wu C, Li B (2007) rstream: resilient and optimal peer-to-peer streaming with rateless codes. IEEE Trans Parallel Distributed Sys 19(1):77–92
48. Yu Z, Wei Y, Ramkumar B, Guan Y (2008) An efficient signature-based scheme for securing network coding against pollution attacks. In: IEEE INFOCOM 2008 - The 27th conference on computer communications, pp. 1409–1417
49. Zhao X (2012) Notes on "exact decoding probability under random linear network coding" IEEE Commun Lett 16(5):720–721

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Rossano Gaeta received his Laurea and Ph.D. degrees in Computer Science from the Università di Torino, Italy, in 1992 and 1997, respectively. He is currently Associate Professor at the Computer Science Department, Università di Torino. He has been recipient of the Best Paper award at the 14th IEEE/ACM MASCOTS 2006 and at the 26th IFIP PERFORMANCE 2007. His current research interests include the analysis of coding techniques in distributed applications and the modeling of information diffusion in online social networks.

Marco Grangetto received his Electrical Engineering degree and Ph.D. degree from the Politecnico di Torino, Italy, in 1999 and 2003, respectively. He is currently Full Professor at the Computer Science Department, Universit'a di Torino. His research interests are in the fields of multimedia signal processing and networking. In particular, his expertise includes wavelets, image and video coding, data compression, video error concealment, error resilient video coding unequal error protection, and joint source channel coding.